CS 6375
ASSIGNMENT 1

Names of students in your group:
Yagna Srinivasa Harsha Annadata (YXA210024)
Vishruth Reddy Chinthi Reddy (VXC220020)

Number of free late days used: 0

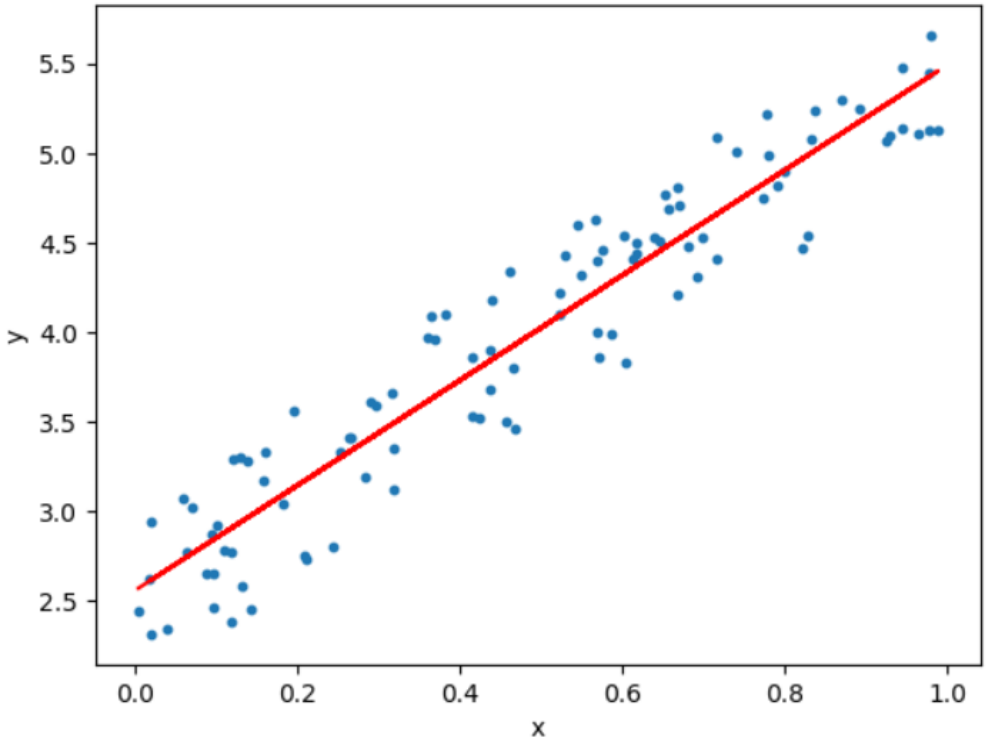
Note: You are allowed a **total** of 4 free late days for the **entire semester**. You can use at most 2 for each assignment. After that, there will be a penalty of 10% for each late day.

Please list clearly all the sources/references that you have used in this assignment.

1. Gradient Descent in Linear Regression - GeeksforGeeks
2. Linear Regression using Python. Linear Regression is usually the first… | by Animesh Agarwal | Towards Data Science
3. https://aegis4048.github.io/mutiple_linear_regression_and_visualization_in_python

Part1: Linear Regression using Gradient Descent

Linear regression is used to predict the result of certain variables based on other dependent variables. The objective of this model is to obtain the most accurate predicted outcome.



A simple linear regression model equation is given as:

$$Y = w_0 + w_1 x_1$$

where Y is predicted value and x is the independent variable.

For the multiple linear regression, equation would be:

$$Y = w_0 + w_1 x_1 + w_2 x_2 + \ldots\ldots\ldots + w_n x_n$$

Here w_i are coefficients of the independent variables x_i and Y is the prediction variable.

Cost Function:

Cost function or Error function is the calculated function of the error between predicted values and actual values. The cost function of a linear regression is root mean squared error.

$$J = \frac{1}{n} \sum_{i=1}^{n} (\text{pred}_i - y_i)^2$$

Gradient Descent:

Gradient descent (GD) is an iterative first-order optimization algorithm that utilizes the first derivative of a function to locate a nearby minimum or maximum point within the function's domain.

Dataset Used:

Wine Quality

Independent Variables:
x1 = fixed acidity
x2 = volatile acidity
x3 = citric acid
x4 = residual sugar
x5 = chlorides
x6 = free sulfur dioxide
x7 = total sulfur dioxide
x8 = density
x9 = pH
x10 = sulphates
x11 = alcohol

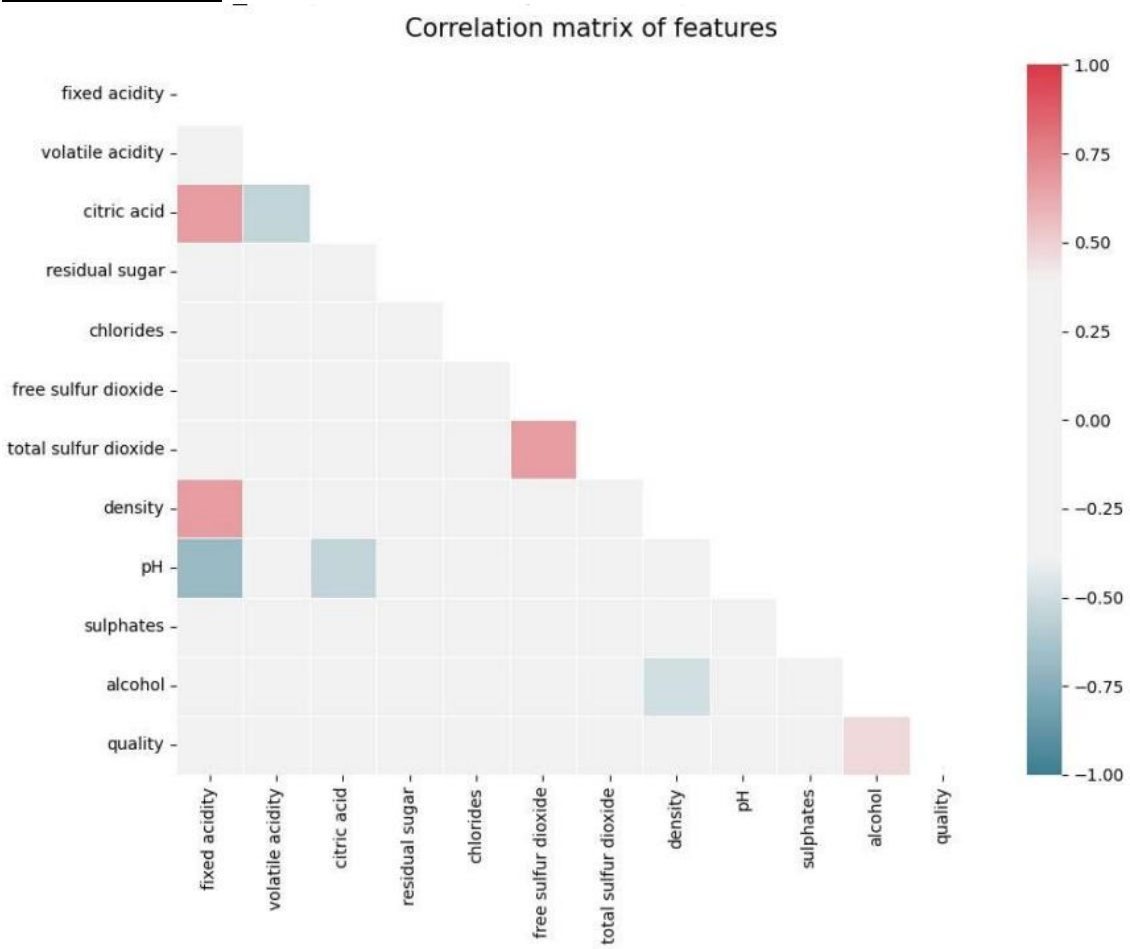
Dependent Variables:
Y = Quality

Problem Definition:

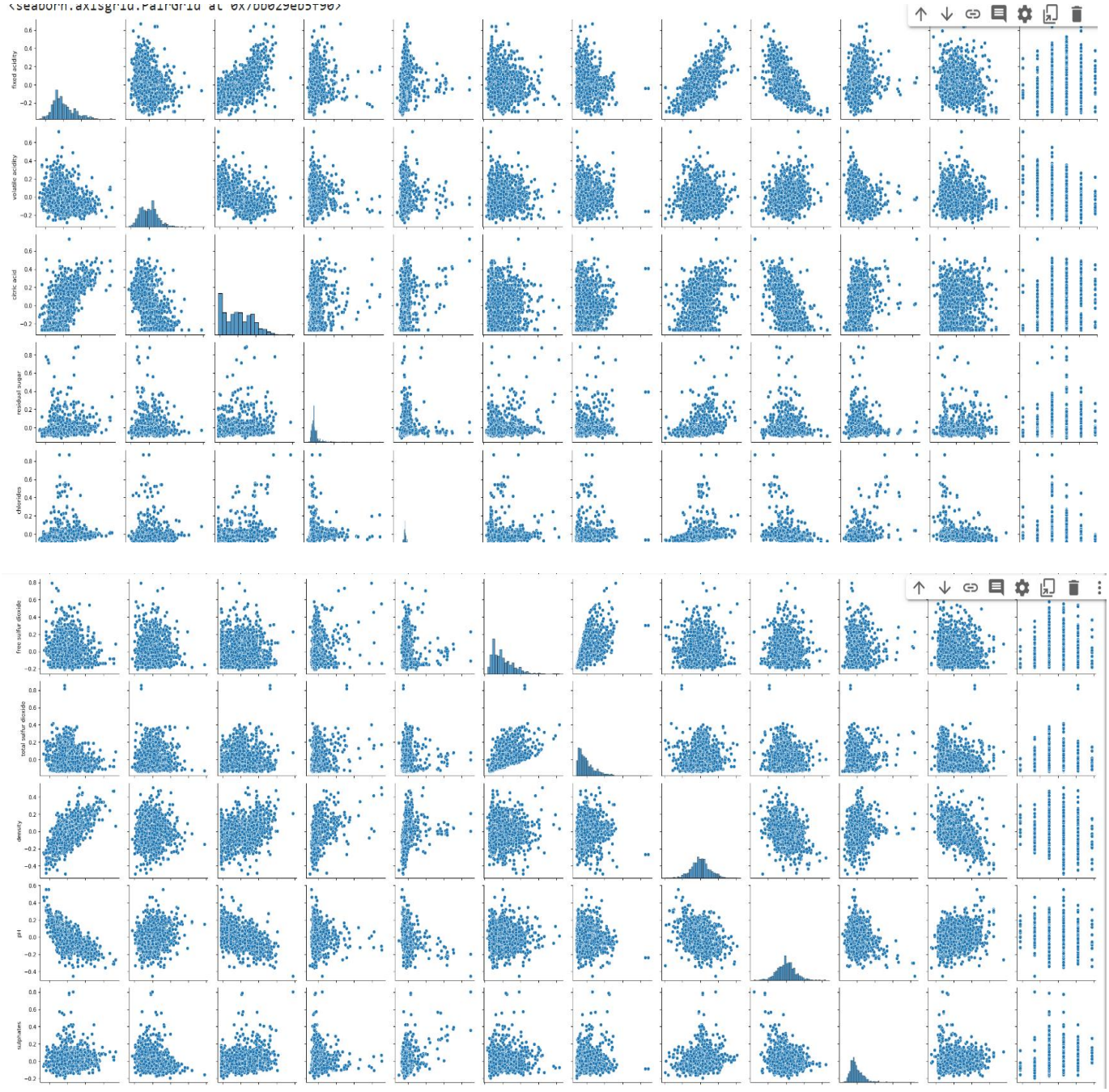
Quality of wine is the most important feature in wine making. The goal is to model wine quality based on physicochemical tests. The features which determine the quality of wine here are Fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol.

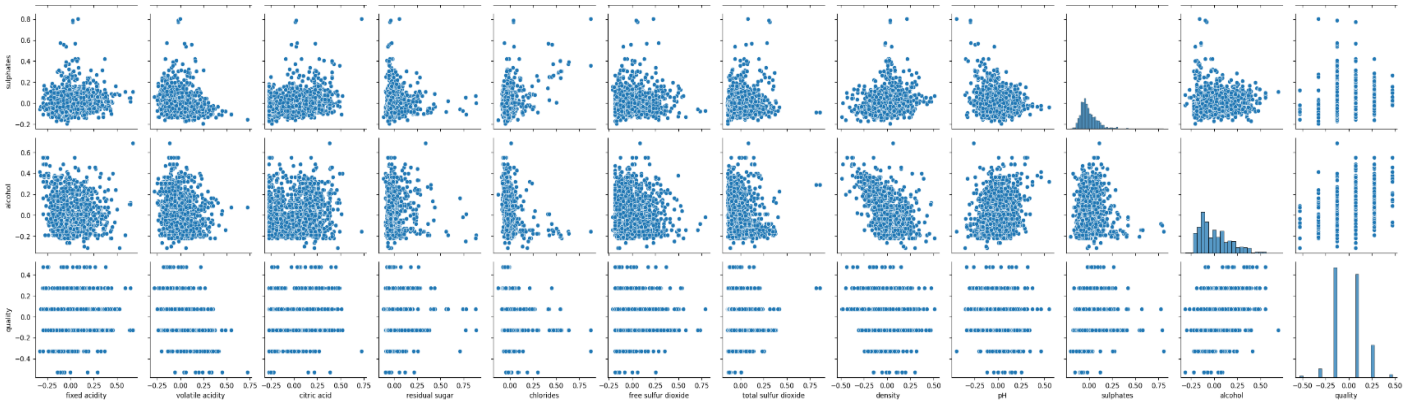
Plots and Observations:

Correlation Plot:

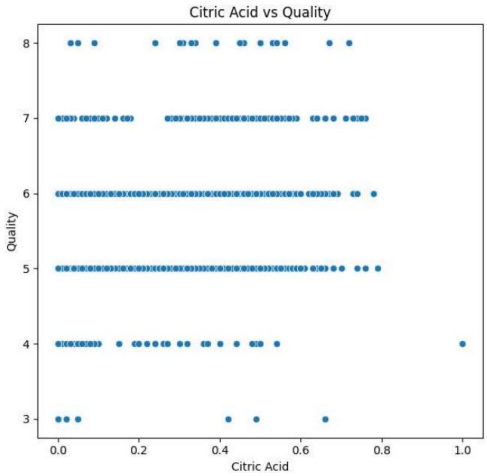


Pair plots with all the variables:

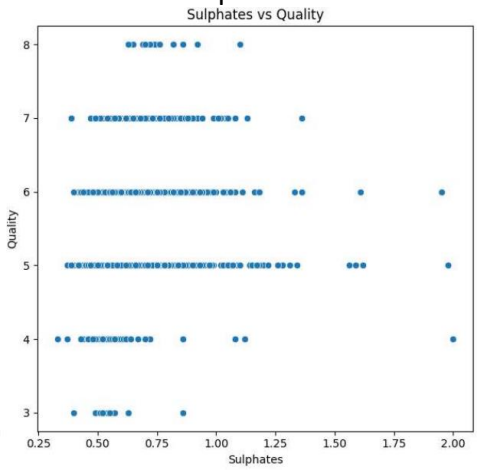




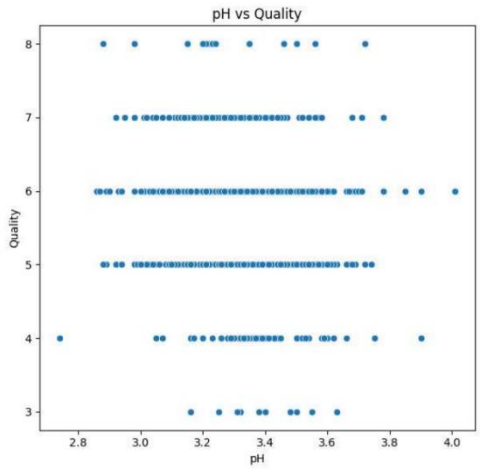
Plot between quality and citric acid:



Plot between sulphates and citric acid:

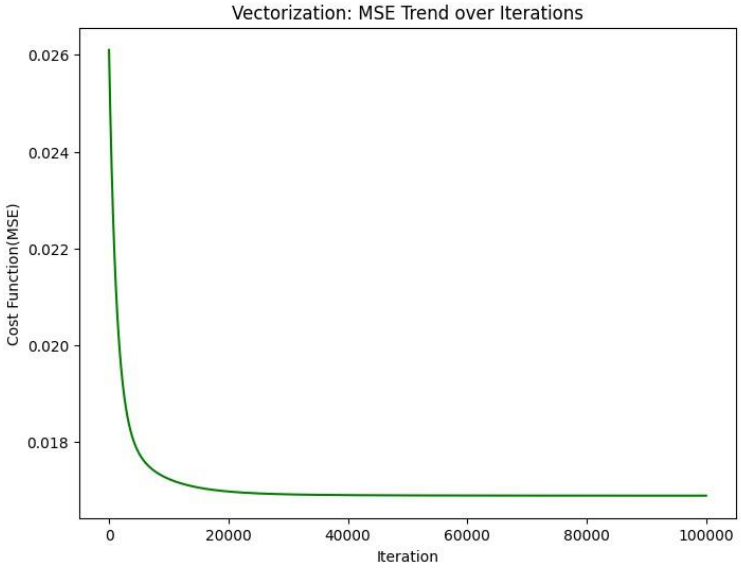


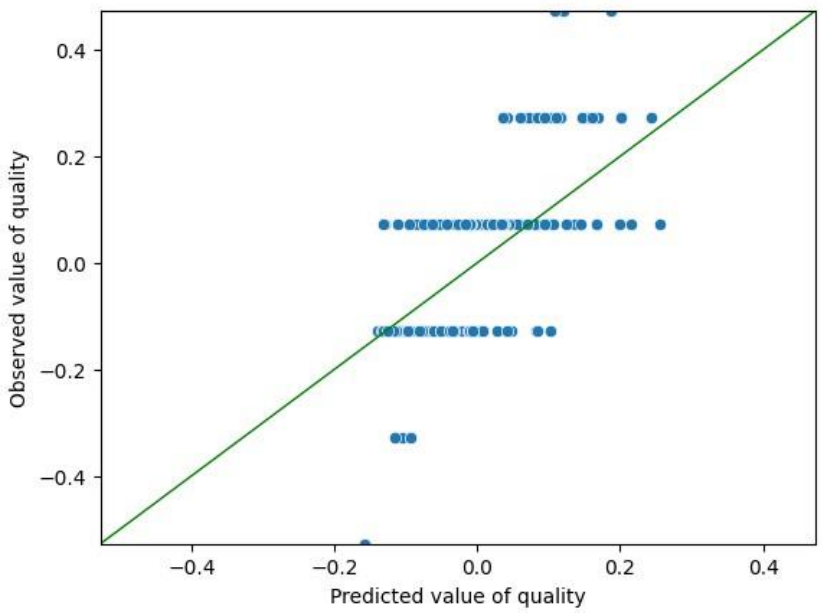
Plot between quality and citric acid:



Part1:
Google Colab link:
https://colab.research.google.com/drive/1ucTWgC4sjZV9zx48V1WzYP_JNPc5Zafs#scrollTo=EpCPEmcYh9id
Dataset Link:
https://raw.githubusercontent.com/YagnaAnnadata/Projects/main/LinearRegression/Dataset/winequality/winequality-red.csv
Original Dataset Link:
https://archive.ics.uci.edu/dataset/186/wine+quality

Observations:





Question: Are you satisfied that you have found the best solution? Explain.

Ans:

- The gradient descent algorithm tries to minimize the error between actual and predicted value of dependent variable. As the number of iterations increases, the cost function minimizes over certain values of theta and then, it again increases.
- Since we have all features which have almost same correlation and considering only one feature in the model doesn't improve the model accuracy, we decided to use all the features which are present in dataset.
- We tried to tune the parameters by setting constant seed. The parameters which we used to tune the model are mentioned below.

	Learning Rate	Number of iterations	Theta	Test Size	Random State	R-Square (test)
1	0.01	50000	np.ones(X_train.shape[1])	0.2	41	0.3149
2	0.01	100000	np.ones(X_train.shape[1]) * X_test.mean()	0.2	41	0.3346
3	0.01	50000	np.ones(X_train.shape[1]) * X_test.mean()	0.1	40	0.4372
4	0.01	100000	np.ones(X_train.shape[1]) * X_test.mean()	0.1	40	0.4394

- We believe our model with learning rate 0.01, number of iterations 100000, test size 0.1, random state: 40 generated the best solution.
  Details of other metrics output are present in log files.

Part-2:
Google Colab link: https://colab.research.google.com/drive/115l2KtNZzn-azpiad2mb-B8zkVdt6jO8#scrollTo=NwhznfVp9ODY

Question: Are you satisfied that the package has found the best solution? How can you check? Explain.

- We have used the scikit-learn library to execute a linear regression model, including the normalization process. The results indicate that it has successfully found the optimal solution.
- The linear regression approach from the scikit-learn linear model library performs exceptionally well with the given dataset. We have achieved the best R-squared value.
- After comparing these outcomes with the results obtained through manual gradient descent implementation, both methods are functioning in a similar manner and producing similar results. Therefore, we are satisfied with the outcomes we found using the library.

Libraries Used:

- NumPy
- Pandas
- Scikit-learn
- Matplotlib
- Seaborn
- Requests

Linear Regression using Gradient Descent algorithm vs ML Library:

Evaluation Metrics	Gradient Descent Algorithm		ML Library Function	
	Training	Testing	Training	Testing
Mean Absolute Error	0.100630	0.097254	0.622701	0.602489
Mean Square Error	0.016891	0.014800	0.647850	0.567885
R2 Error	0.350673	0.439401	0.350733	0.439205